

Chapter 12: Advanced Issues in Corporate and Acquisition Modelling Involving Resolving Circular References Associated with Interest Expense and Debt Repayments

Circular References in Corporate and Acquisition Models

One issue that not too many people in the world care about is coming up with an elegant way to compute the interest expense on the average of opening and closing debt balance rather than simply the opening balance in financial models without using the iteration button in excel and without using a macro. This is an important issue to some people because in a model with items computed on an annual basis, cash flow after interest expense affects debt balance, but the debt balance affects interest expense which in turn affects cash flow. While the problem of more accurately measuring interest expense may be trivial to most people and is certainly not a key driver of virtually any analysis (it is fine to use the opening balance in general), solving the problem illustrates elegant ways in which functions can be used to solve problems in financial models. For many advanced modelling issues, creating your own user defined functions opens a whole new world of solving problems that otherwise use things like tedious copy and paste macros.

Some of the techniques that can be used to solve circular references such as the problem of interest expense on the average debt balance are listed below. The table shows five different techniques along with comments regarding the problems with the method. Because there are big problems with the macro iteration button method and the copy and paste macro method, this chapter describes how to create a flexible and transparent function using one of the last two methods -- the algebraic approach or the isolated function.

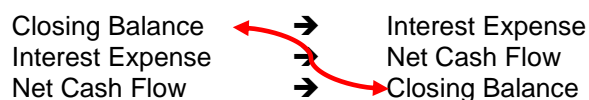
Method	Problems with Method
Use the "Enable Iterative Calculation" button in Excel (the iteration button)	This method can destroy models and is dangerous to use. The models become unstable, goal seek does not work and errors cannot be undone. The method should virtually never be used.
Input fixed values in cells that cause the circular problem and use a goal seek to find the fixed value by setting the difference to zero	This method can avoid the dangers of the iteration button but it means that whenever you must change something in a model, you must run a goal seek. Another problem with this method is that it does not solve multiple circular reference problems.
Find the value of fixed values in a similar manner to goal seek method, but use the solver to find multiple values at the same time.	Using the solver resolves problems of the goal seek whereby multiple circular references can be established. But it leaves the problem that you must re-run the solver any time you change something. Another big problem is the solver can be very slow and clumsy for anything but very small models. It is also limiting because all calculations must be in one sheet.
Use a copy and paste macro to find fixed values with an iteration routine.	This is the most common method used these days for fancy models. You compute the item causing the circular reference and they copy and paste special to a fixed cell. The method solves the problem with goal seek where multiple circular references cannot be solved and it is better than the solver method which can be very slow and limits the calculations to one sheet. But this method leaves the models just as clumsy as the above method and it means the copy and paste macro must be used whenever you want to do any sensitivity analysis or optimisation.
Resolve circular references using an algebraic equation	This is an old fashion method that is the most elegant way to solve circular references. It requires no iteration on an indirect or a direct basis and all optimisation and scenario analysis can be directly applied. The problem with this method is that coming up with an algebraic solution can be tedious and sometimes almost impossible.
Resolving circular references using an isolated function that contains an iteration loop	This method in which calculations that cause the circular reference are repeated in a function resolves problems with the clumsiness and the untransparency caused by standard copy and paste macros. All data tables, optimisation and scenario analysis can be automated and any circularity reference can be resolved. The only problem with this method is that it involves a bit of old fashion programming.

Step One in Resolving Circular References: Understanding Why Circular References Arise and Whether they can be Solved

In describing how to solve the problem of circular references and other challenging problems in modelling, it is useful to begin with a simple problem and gradually add complexities until you get completely stuck. The organisation of this chapter follows such an approach in describing how to resolve the circular reference problem in models other than project finance models. An acquisition model with a cash flow waterfall without taxes and a cash flow sweep is the starting point for the analysis. This case is relatively simple because there is generally no difference in interest rates between surplus cash earnings and interest expense. Further, in the first example, there is no effect of the interest expense on taxes and on dividend payouts. Taxes (particularly with net operating losses), differences between interest income and interest expense rates and dividends (especially with

limits when earnings are negative) complicate the problem. In demonstrating how to solve this problem, each of the five methods summarised above is illustrated.

Corporate models and acquisition annual models are often annual models. In these models an implicit assumption is that the cash flows occur on a continual basis throughout the year. When computing various items it is reasonable to approximate the continuing cash flow with an assumption that cash flow occurs in the middle of the period. Given this notion, it is not reasonable to assume that incremental interest expense or incremental interest income can be computed on the basis of the opening debt or cash balance. If the opening balance is used for interest expense on new debt or interest income on new surplus cash generated, the implicit assumption is that all revenues received and all expenses and capital expenditures happen at the end of the year. If the opening balance of debt is used in calculations meaning that the balance of debt does not change until the end of the period, then the interest expense is not precise. The true interest expense would be less than interest computed on the opening debt balance if cash flow were used to repay debt over time. Making an assumption that the revenues occur in the middle of the period which is much more reasonable creates a circular reference because the closing balance of the debt or cash affects the interest expense. But the interest expense is a determinant of cash flow that drives the closing balance as illustrated below:



Some people suggest that from a philosophical standpoint that circularity should not occur in the models because it does not occur in real situations. For example, if circularity existed in financing a transaction, the process would work something like the following. The sponsor would go to the bank and ask for a loan commitment. Then, the bank uses the amount of the loan commitment to compute fees. After that, the sponsor asks for a larger loan to cover the fees that were not known before he asked the bank for the loan. With the larger loan, more fees are charged and the sponsor needs an even higher loan. Then the loan is higher and process keeps going on and on and on. The circular reference resembles the film "Groundhog Day" where life cannot move forward and the same problems occur again and again and again. Instead of the above scenario with the banker and the sponsor going around, it is more realistic if the sponsor asks for a loan commitment that will already include the fees, the debt service reserve account, and the interest during construction that can cause a circular reference problem. When a company goes to a bank to ask for a loan in a project financing transaction, the company has already created a model and evaluated the sizing of debt, presumably through modelling the debt service coverage or some other criteria. This is the philosophy that circular references should not be in a model. Notwithstanding philosophical considerations about whether circular references occur in the real world, various combinations of the funding approaches cannot be solved without running into a circular reference problem where the debt or equity commitment drives the total funding needs and when the funding needs drive the financing commitment. From a modelling standpoint, the fees and the loan commitment depend on the financial model, but the model must be computed with some kind of fee and debt commitment assumption. When an item in the model depends on using the model itself, a circular reference sometimes cannot be avoided.

Circular References from a Cash Flow Sweep using the Iteration Button, Goal Seek, Solver and Copy and Paste Macros

To illustrate the problem of circular references in non-project finance models, you can create a simple example with a cash flow sweep where interest expense is computed on the basis of the average debt balance rather than the opening debt balance. In the cash sweep example, the cash flow before financing is given and the cash flow available for re-paying debt is the minimum of the cash after interest or the opening balance:

$$\text{Debt repayment} = \text{MIN}(\text{cash flow after interest}, \text{opening debt balance})$$

You can create a debt schedule below the cash flow analysis with an opening balance, a repayment from the formula above, a closing balance and an interest expense calculation. Because the interest expense is computed on the basis of the average balance, a component of interest depends on debt

repayment which is in turn dependent on the cash flow after interest as illustrated by splitting the interest expense formula:

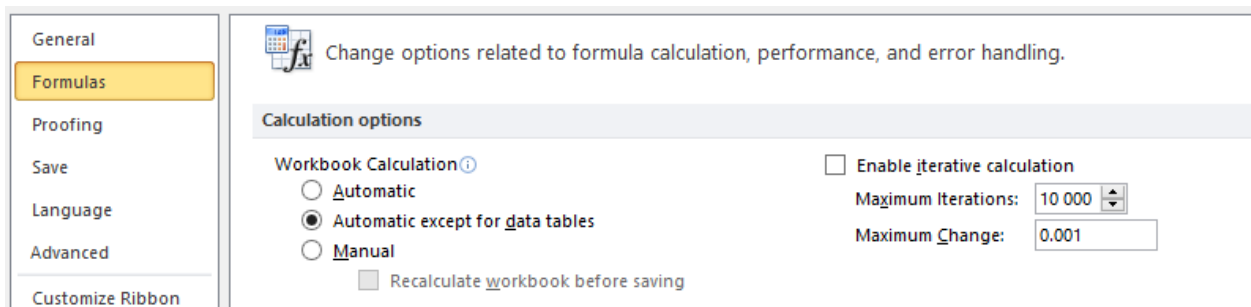
$$\text{Interest Expense} = \text{Beginning Balance}/2 \times \text{Rate} + \text{Ending Balance}/2 \times \text{Rate}$$

Since the ending balance is the beginning balance less the cash sweep, the interest expense is a function of debt repayment:

$$\text{Interest Expense} = \text{Beginning Ending Balance}/2 \times \text{Rate} + (\text{Beginning Balance} - \text{Sweep})/2 \times \text{Rate}$$

If you make this little model, then after you calculate the interest expense a dreaded blue arrow will appear indicating you have a circular reference. Excel has an option to resolve this kind of circular reference through making iterative calculations using a special button. This button seems to make the whole circular reference issue go away. The iteration button can be accessed by going to the File tab, clicking on Formulas and then selecting the Enable iterative calculation check box as illustrated below. This solution seems to be a nice feature that eliminates the problem. But it can be a very dangerous thing to do. Some of the negative consequences of using the iteration button include:

- (1) the model can become unstable meaning that when you make an error that results in a bunch of #VALUE's, you cannot use the undo key. This can be a real pain;
- (2) the goal seek tool and some other tools do not work;
- (3) if the model is large, it can become very slow as the whole model will be computed over and over again; and,
- (4) sometimes the model does not even work;



The next few paragraphs discuss alternative ways to resolve the circular references without using the iteration button. While these methods correct the stability problems, the alternative solutions can create new problems and be worse than the using the iteration button.

One solution to the problem involves re-computing the interest expense somewhere outside of the cash flow process. After you make this calculation, you can enter a fixed value for the interest expense in the body of the model. The difference between the computed value and the fixed value should be zero. To compute the fixed value you can use the GOALSEEK tool or the SOLVER tool as follows:

- Step 1: Enter a fixed value for the interest expense computed in the cash flow statement
- Step 2: Compute the difference between the fixed interest expense and the computed interest expense below the calculation of interest expense on the average balance in the debt schedule
- Step 3: Use the GOALSEEK tool to set the difference to zero by finding the fixed interest expense amount.

Step 4: Repeat the process for each year of the model

Instead of repeating the goal seek in step 4, you could use the SOLVER tool. In this case, do not put anything in the top part, set the changing cells to the series of fixed interest expenses and add a constraint that the difference must be zero. Either the GOALSEEK or the SOLVER method are have many problems as follows:

- (1) If you have another circular reference somewhere else, the process can be difficult (it may be impossible with the GOALSEEK).
- (2) If you want to use the GOALSEEK tool to do things like find debt capacity or break-even points, the process can become very cumbersome.
- (3) You cannot use the DATA TABLE and other tools.

An approach that is analogous to the GOALSEEK method is to create a copy and paste macro. Some people consider themselves sophisticated if they can apply this method because they are creating a macro. In general this approach can be constructed using the following steps:

Step 1: Make sure there is a fixed value and a computed value for the thing that is causing the circularity as described above (e.g. the average interest expense calculated in the debt schedule).

Step 2: Compute the year by year difference between the two calculations and then sum all of the differences in a separate cell.

Step 3: Begin recording a macro. Then copy and paste special as values from the computed interest expense to the fixed interest expense. Create range names for the computed interest expense, the fixed interest expense and the sum of the difference in interest expense cells.

Step 4: Modify the macro by including range names and including a while loop. The while loop repeats the copy and paste command until the sum of the differences converges to zero. You can be fancier with the copy and paste macro, but this is the essence of the process. Making a macro that includes a while loop is illustrated below.

```
Sub CopyPaste()  
'  
' CopyPaste Macro  
'  
While Range("difference") <> 0  
'  
    Range("computed").Select  
    Selection.Copy  
    Range("fixed").Select  
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _  
        :=False, Transpose:=False  
  
Wend  
  
Application.CutCopyMode = False  
  
End Sub
```

Fancier ways to add bells and whistles to copy and paste macros (adding iteration limits, displaying the iterations and simplifying the code) are described below in the context of project finance analysis.

The copy and paste method has many serious advantages even if you attach the macro to a fancy button and call it an optimisation routine. With a copy and paste macro, the model cannot be used together with the goal seek or solver tools; data tables cannot be created for sensitivity analysis; the iteration can occasionally result in an infinite loop; and doing any sort of analysis with the model can become a cumbersome ugly process.

Solving Basic Circular References in with Cash Sweeps in Acquisition Models using Algebra

A more elegant solution to the problem of circular references than using macros or allowing the iteration button to be active is to create a function that derives the interest expense from an algebraic expression. The algebraic solution is complicated by taxes, dividends and variations in interest rates between income and expense. To begin the discussion, consider an acquisition model where all excess cash flow is used to pay down debt before any dividends are allowed (a cash flow sweep). Further assume no taxes. In this case, the repayment of debt can be computed in the cash flow statement using the MIN function consistent with the discussion of cash flow waterfalls above.

When working through the circular reference functions through creating an algebraic process, it is generally best to solve for the repayment and write down a bunch of equations that include a term for debt repayment. You can then substitute variables, re-arrange things and ultimately replace the cash flow after interest with an equation that does not depend on repayment. This equation for cash flow that does not depend on debt repayment can be used in the cash flow section of the model. In the simple example of a cash flow sweep with no taxes, the arguments for the function are the operating cash flow (EBITDA less working capital changes and capital expenditures.)

To solve the circular problem, begin by writing down a couple of equations that define interest expense and the repayment of debt assuming that the interest is derived from the average balance rather than the opening balance:

$$\text{Interest Expense} = \text{Opening Balance} \times \text{Interest Rate} - \text{Repayment} \times \text{Interest Rate} / 2$$

$$\text{Repayment} = \text{MIN}(\text{Cash flow}, \text{Opening Balance})$$

$$\text{Cash Flow} = \text{Operating Cash Flow} - \text{Interest Expense}$$

Using these equations, you can create an expression for the repayment by substituting and re-arranging variables. In general the algebraic method involves finding an equation for repayment and replace the cash flow equation with a repayment equation as illustrated below:

$$\text{Repayment} = \text{Operating Cash Flow} - (\text{Opening Balance} \times \text{Interest Rate} - \text{Repayment} \times \text{Interest Rate} / 2)$$

$$\text{Repayment} + \text{Repayment} \times \text{Interest Rate} / 2 = \text{Operating Cash Flow} - \text{Opening Balance} \times \text{Interest Rate}$$

$$\text{Repayment} \times (1 + \text{Interest Rate} / 2) = \text{Operating Cash Flow} - \text{Opening Balance} \times \text{Interest Rate}$$

$$\text{Repayment} = (\text{Operating Cash Flow} - \text{Opening Balance} \times \text{Interest Rate}) / (1 + \text{Interest Rate} / 2)$$

Once you have found an equation it is sometimes fairly long and painful to put into excel. As an alternative, you can put the equation into a function. Then you can combine your function (which could be named SWEEP) with the MIN function as illustrated below:

$$\text{Repayment} = \text{MIN}(\text{SWEEP}(\text{operating cash flow}, \text{interest rate}), \text{Opening Balance of Debt})$$

Contents of the function and implementation of the function in the model are illustrated in the diagrams below. Using this function avoids all of the problems associated with the iteration button, the goal seek and the copy and paste method. The model is stable and does not have to make iterations; you can run goal seeks, solvers and data tables; the model does not slow down and the model is transparent.

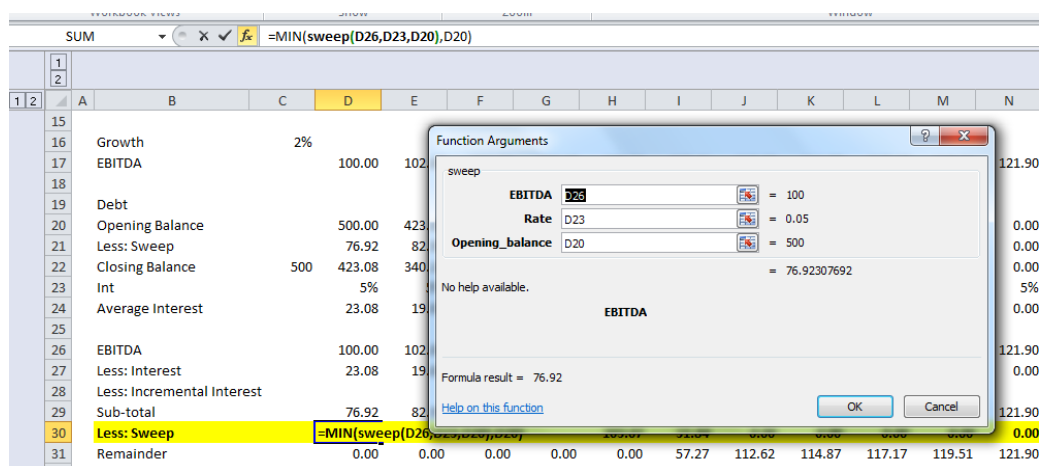
Function sweep(EBITDA, rate, Opening_balance)

$$\text{sweep} = (\text{EBITDA} - \text{Opening_balance} \times \text{rate}) / (1 - \text{rate} / 2)$$

End Function

Growth	2%											
EBITDA		100.00	102.00	104.04	106.12	108.24	110.41	112.62	114.87	117.17	119.51	121.90
Debt												
Opening Balance		500.00	423.08	340.16	250.89	154.92	51.84	0.00	0.00	0.00	0.00	0.00
Less: Sweep		76.92	82.92	89.26	95.98	103.07	51.84	0.00	0.00	0.00	0.00	0.00
Closing Balance	500	423.08	340.16	250.89	154.92	51.84	0.00	0.00	0.00	0.00	0.00	0.00
Int		5%	5%	5%	5%	5%	5%	5%	5%	5%	5%	5%
Average Interest		23.08	19.08	14.78	10.15	5.17	1.30	0.00	0.00	0.00	0.00	0.00
EBITDA		100.00	102.00	104.04	106.12	108.24	110.41	112.62	114.87	117.17	119.51	121.90
Less: Interest		23.08	19.08	14.78	10.15	5.17	1.30	0.00	0.00	0.00	0.00	0.00
Less: Incremental Interest												
Sub-total		76.92	82.92	89.26	95.98	103.07	109.11	112.62	114.87	117.17	119.51	121.90
Less: Sweep		=MIN(sweep(D26,D23,D20),D20)										
Remainder		0.00	0.00	0.00	0.00	0.00	57.27	112.62	114.87	117.17	119.51	121.90

In entering the function, you can use the f_x box to guide the inputs to put into the formula as illustrated below:



The problem with the algebraic method is that equations become more tedious with more complex problems. But the principle of substituting variables and re-arranging equations is the same. The bad news about more tedious formulas is that they are painful and sometimes boring to work through. The good news is that if you create a function and use variable names that are easy to interpret, you only have to do it one time. Once you have made the function, you can copy the function to other functions or even create an add-in to excel. The formulas are generally easy to transfer because by the time you get down to an income statement and a cash flow statement in a financial model, the format should be almost identical.

Solving Circular References with Functions that Repeat the Equations that Cause the Problem

The final method for solving the circular reference problem extends the function method introduced above and applies some of the iteration ideas from the prior methods. The final approach solves problems in a similar manner as the algebraic method, but you do not have to work through subtitled the equations. Creating a function with an iteration process allows you to use the GOAL SEEK, DATA TABLES and keeps the model stable. This method often has advantages relative to algebraic method because finding a single equation is very difficult if not impossible in many circular reference problems such as the case with taxes and a net operating loss.

To apply the method of creating a function the first step is to decide which equation should be fixed with the function so as to remove the circular reference. Next you can write down equations for the items that are related to the equation that is causing the circular reference. Once the equations are

established, all of the inputs to the equations must be included as inputs to the function (you cannot use cell references when writing equations in a function. Finally, an iteration process should be added to the function where the equations are repeated until some kind of convergence is established. For the case of the cash flow sweep, the function could be defined for the interest expense. The interest expense depends on the average debt and the cash flow. But the cash flow depends on the interest expense. The iteration can compare the interest expense from the prior iteration with the currently computed interest expense. The function below illustrates the process. The last step in the function is testing whether the sweep computed from the prior iteration is the same as the sweep computed from the same iteration. If the last iteration and the current iteration are the same, the iteration process can be stopped and the function can be stopped.

```

Function average_interest(interest_rate, cash_flow, opening_debt_balance)

For Iteration = 1 To 30
    last_sweep = sweep

    average_interest = ((opening_debt_balance + closing_debt_balance) / 2) * interest_rate

    sweep = WorksheetFunction.Min(cash_flow - average_interest, opening_debt_balance)

    closing_debt_balance = opening_debt_balance - sweep

Next Iteration

converge = Abs(sweep - last_sweep)

If converge < 0.000001 Then Exit Function

End Function

```

This function works in the same way as the algebraic technique above in that the file does not lose any functionality associated with file stability, sensitivity analysis or optimisation. Further, the function is fairly easy to change with more complex equations. When taxes are included in the analysis, computing the equation using the algebraic method becomes more complex. The problem with including taxes in a financial model with circularity is that interest expense affects taxes that in turn affects the cash flow available to repay debt. Because the taxes are part of the interest calculation, an equation for taxes must be included in the function as shown below. This example of adding taxes demonstrates that the equations in the function must correspond to the equations in the model. If equations for a net operating loss carry forward were also included in the cash flow analysis, using the algebraic equation would not be possible because of the conditional nature of the operating loss and because of the accumulation of the carryover. However, if the function is used the net operating loss can be incorporated. To include the loss carry forward, the function must read in the opening carry forward balance as well as the tax rate. Then the carry forward can be included in the function using the same equations that are in the excel model as illustrated below.


```

Function average_interest_taxes(interest_rate, cash_flow, opening_debt_balance, tax_rate, opening_NOL_Balance)

For Iteration = 1 To 30
    last_sweep = sweep

    average_interest_taxes = ((opening_debt_balance + closing_debt_balance) / 2) * interest_rate

    taxable_income = cash_flow - average_interest_taxes
    NOL_created = WorksheetFunction.Max(0, 0 - taxable_income)
    NOL_used = WorksheetFunction.Min(opening_NOL_Balance, WorksheetFunction.Max(0, taxable_income))
    closing_NOL_Balance = opening_NOL_Balance + NOL_created - NOL_used
    adjusted_taxable_income = taxable_income + NOL_created - NOL_used
    taxes_paid = adjusted_taxable_income * tax_rate

    sweep = WorksheetFunction.Min(cash_flow - average_interest_taxes - taxes_paid, opening_debt_balance)

    closing_debt_balance = opening_debt_balance - sweep

Next Iteration

converge = Abs(sweep - last_sweep)

If converge < 0.000001 Then Exit Function

End Function

```

Operating the function involves using the f_x thing in the excel formula box as illustrated below:

The screenshot shows an Excel spreadsheet with a 'Function Arguments' dialog box open. The dialog box is for the 'average_interest_taxes' function, which is defined in the code block above. The arguments are as follows:

Argument	Value	Result
Interest_rate	0.2	
Cash_flow	100	
Opening_debt_balance	600	
Tax_rate	0.3	
Opening_NOL_Balance	0	
		122.222222

The formula result is 122.22. The spreadsheet shows a 'Debt Schedule' table with columns for time and values, and a 'Cash Flow Statement' table with columns for time and values.

Time	Opening Balance	Less: Repayment from Sweep	Closing Balance	Interest Expense on Average Balance	Check
1	600.0	-22.2	600.0	122.2	124.94
2	600.0	-22.2	600.0	122.2	124.94
3	600.0	-22.2	600.0	122.2	124.94
4	600.0	-22.2	600.0	122.2	124.94
5	600.0	-22.2	600.0	122.2	124.94
6	600.0	-22.2	600.0	122.2	124.94
7	600.0	-22.2	600.0	122.2	124.94
8	600.0	-22.2	600.0	122.2	124.94
9	600.0	-22.2	600.0	122.2	124.94
10	600.0	-22.2	600.0	122.2	124.94
11	600.0	-22.2	600.0	122.2	124.94
12	600.0	-22.2	600.0	122.2	124.94
13	600.0	-22.2	600.0	122.2	124.94
14	600.0	-22.2	600.0	122.2	124.94
15	600.0	-22.2	600.0	122.2	124.94
16	600.0	-22.2	600.0	122.2	124.94
17	600.0	-22.2	600.0	122.2	124.94
18	600.0	-22.2	600.0	122.2	124.94
19	600.0	-22.2	600.0	122.2	124.94
20	600.0	-22.2	600.0	122.2	124.94
21	600.0	-22.2	600.0	122.2	124.94
22	600.0	-22.2	600.0	122.2	124.94
23	600.0	-22.2	600.0	122.2	124.94
24	600.0	-22.2	600.0	122.2	124.94
25	600.0	-22.2	600.0	122.2	124.94
26	600.0	-22.2	600.0	122.2	124.94

Time	Cash Flow	Less: Interest Expense	EBT before NOL	NOL Balance	Opening Balance
1	100.00	122.22	-22.22	0.00	22.22
2	100.00	122.22	-22.22	0.00	22.22
3	100.00	122.22	-22.22	0.00	22.22
4	100.00	122.22	-22.22	0.00	22.22
5	100.00	122.22	-22.22	0.00	22.22
6	100.00	122.22	-22.22	0.00	22.22
7	100.00	122.22	-22.22	0.00	22.22
8	100.00	122.22	-22.22	0.00	22.22
9	100.00	122.22	-22.22	0.00	22.22
10	100.00	122.22	-22.22	0.00	22.22
11	100.00	122.22	-22.22	0.00	22.22
12	100.00	122.22	-22.22	0.00	22.22
13	100.00	122.22	-22.22	0.00	22.22
14	100.00	122.22	-22.22	0.00	22.22
15	100.00	122.22	-22.22	0.00	22.22
16	100.00	122.22	-22.22	0.00	22.22
17	100.00	122.22	-22.22	0.00	22.22
18	100.00	122.22	-22.22	0.00	22.22
19	100.00	122.22	-22.22	0.00	22.22
20	100.00	122.22	-22.22	0.00	22.22
21	100.00	122.22	-22.22	0.00	22.22
22	100.00	122.22	-22.22	0.00	22.22
23	100.00	122.22	-22.22	0.00	22.22
24	100.00	122.22	-22.22	0.00	22.22
25	100.00	122.22	-22.22	0.00	22.22
26	100.00	122.22	-22.22	0.00	22.22